iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: I Month of publication: January 2025

DOI: https://doi.org/10.22214/ijraset.2025.66705

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue I Jan 2025- Available at www.ijraset.com

Review of Large Language Models for Geospatial Query Understanding and Generation

Deepali Ahir[1], Shishir N[2], Kaushal B.[3], Aditya S.[4], Sunil Kumawat[5]
Department of Computer Engineering, MES Wadia COE, SPPU, Pune, 411001

Abstract: This paper examines Large Language Models (LLMs) for the task of interpreting and generating natural language queries, with a particular focus on the geographic data domain. This paper examines three primary functions: Text-to-SQL, which converts natural language queries to SQL queries for relational databases, Text-to-OverpassQL and Corpus query language, which enters the language analysis. This review examines various approaches used to improve the performance of Large Language Models (LLMs) in these areas, including rapid development, optimization, and data augmentation techniques. It also addresses issues such as the ambiguity of natural language, the complexity of geographic data, the need for specialized knowledge, and the visual problem. This paper describes in more detail the underlying data and measurement techniques used in this area. By tying together existing research, this paper highlights the potential of LLM to provide independent access to geographic data and identifies avenues for future research, such as improving security, integrating external information, and improving more comprehensive measurement.

Keywords: Large Language Models, Text-to-SQL, Text-to-OverpassQL, OpenStreetMaps (OSM), Geospatial Analysis, NLP, GIS (Geographic Information System)

I. INTRODUCTION

Geospatial analysis plays an important role in many disciplines, including urban planning, transportation, environmental studies, and disaster management. This leads to an understanding of the relationships between areas, structures, and patterns that are essential for informed decision making in these and other areas. For example, geospatial analysis is vital for applications such as supply chain management, disaster response, and public health assessments. Despite their importance, traditional geographic information systems (GIS) are often problematic for inexperienced users.

These systems often involve complex software, specialized data, and require in-depth knowledge of spatial data and analysis, limiting their widespread use. Additionally, traditional GISs often lack the ability to effectively perform natural language processing (NLP) to solve geospatial questions, adding complexity to the user experience. , offers a flexible approach to bridge this gap. LLM eliminates the need for expertise in geospatial programming or tools by allowing users to interact with geospatial data through natural language queries. It provides free access to geospatial data and analytics by translating language queries into executable code or database queries. It focuses on how to use LLM in content analysis to understand user intent and how to use LLM in markup to perform spatial analysis.

We will review the methods used, such as the rapid development and optimization of law graduate programs, and discuss current issues and limitations in this area. Special emphasis will be placed on using LLM to perform tasks such as generating SQL queries, generating OverpassQL for OpenStreetMap (OSM) data, generating CQL expressions for linguistic research, and developing Python code for GIS tasks. This project can make geospatial analysis faster, more efficient, and more effective by improving the accessibility and usability of Geographic Information Systems through LLM. This will help support better decision making across all tasks and enable inexperienced users to benefit from geospatial intelligence. Additionally, LLM research in this area may offer new ways to process and analyze spatial data with implications for areas ranging from urban planning to environmental protection. The following sections will discuss the development of text-to-spatial query engines, the application of LLM to geographic query understanding and generation, and the challenges and opportunities presented by emerging platforms.

II. BACKGROUND

The development of natural language interface for geospatial databases has a rich history, with early systems laying the groundwork for the current focus on Large Language Models (LLMs) in the geospatial domain. Initial goal of these systems was to enable users to interact with databases using natural language rather than technical query languages such as SQL.

A. Evolution of natural language interfaces for databases

The field has moved from manually crafted rules and templates to sophisticated machine learning and deep learning approaches, and most recently to the integration of LLMs. This progression reflects a need to better bridge the gap between human language and structured data. Early approaches primarily focused on pattern matching between natural language and SQL statements, using machine learning models to acquire the mapping between the two. However, the introduction of LLMs has brought a substantial transformation to the field.

B. Early systems and their limitations

One of the earliest attempts to build a natural language interface for geographical data was the **GEOQUERY** system. Based on Prolog and later adapted to SQL, it was tailored for a small database of U.S. geographical facts. While it demonstrated the potential of semantic parsing, it was limited by its specific domain and the need for manual feature engineering. Other rule-based systems, like **LUNAR** and **NaLIX**, also showed the potential of semantic parsing, but they required significant manual effort and were not scalable or adaptable to different domains or complex queries. These early systems often used restricted query languages and small datasets, struggled to handle complex queries with ambiguity and nested structures and were limited by their focus on specific domains. They also required significant manual feature engineering.

C. Progression to more complex systems

As databases and user queries became more complex, systems evolved from rule based approaches to more data-driven methods using neural networks. NLmaps was a more recent effort, which aimed to build a natural language interface for OpenStreetMap (OSM). It introduced a machine-readable language (MRL) as an abstraction of the Overpass Query Language, but it supported only a limited number of OverpassQL features. NLmaps v2 was subsequently released with augmented text and query pairs to facilitate building more potent neural sequence-to-sequence parsers.

D. Current focus on LLMs in geospatial domain:

The current focus is on leveraging the power of LLMs for more complex and versatile applications in the geospatial domain. LLM-based approaches, implementing text-to-SQL through in-context learning and fine-tuning paradigms, have achieved state-of-the-art accuracy. LLMs are now being used for more complex and versatile applications in the geospatial domain and for various geospatial tasks. The use of LLMs has transformed the field by allowing for a more intuitive interface with database systems via natural language.

III.LLMS FOR GEOSPATIAL QUERY UNDERSTANDING

Large language models (LLMs) are increasingly being used to interpret natural language queries related to geospatial data, demonstrating significant promise in areas such as semantic parsing and contextual understanding. These models reduce the gap between human language and the specific requirements of geographic information systems (GIS), making geospatial data more accessible.

A. Semantic parsing of natural language to machine-readable queries

Large Language Models (LLMs) are transforming how we interact with complex systems by enabling semantic parsing—converting natural language into structured, machine-readable formats that GIS tools or databases can understand and act on. This process is vital for translating user requests into actionable commands, such as identifying key components like entities, conditions, and spatial relationships. For geospatial queries, LLMs can translate a simple user request into formats that systems like GIS or databases can execute. For example, asking, "What is the population of London?" might prompt an LLM to generate an accurate SQL query by understanding the user’s intent, the database schema, and the relationships between tables and columns. Some systems also leverage knowledge graphs to improve accuracy when generating SQL or other query languages like OverpassQL for OpenStreetMap or CQL for linguistic analysis. Additionally, LLMs can translate natural language into formats for specific geospatial tools, often using an intermediate step like Machine Readable Language (MRL). For instance, systems like NLmaps convert natural language into MRL, which abstracts OverpassQL, to query OpenStreetMap data. Although MRL supports fewer features, it simplifies query processing and execution while keeping the system user-friendly. The choice of approach, whether generating SQL directly or using MRL, depends on the complexity of the task, the target database, and the system’s goals. By bridging the gap between natural language and technical systems, LLMs make it easier to access and act on complex data.

B. Contextual understanding in geospatial queries

Large Language Models (LLMs) are exceptional at interpreting queries involving location, spatial relationships, and time, making them powerful tools for geospatial analysis. They can translate spatial terms like "near," "within," or "north of" into measurable distances using geospatial tools, handle complex relationships such as adjacency or containment, and interpret vague terms like "close to," often offering users the flexibility to fine-tune parameters. LLMs are particularly skilled at identifying implied locations in queries—such as deducing the meaning of “there” in “What’s the weather like there?”—by leveraging context, user preferences, or previous interactions. They can also process explicit geographical coordinates like latitude and longitude, although linking these to specific real-world locations can sometimes be a challenge. Additionally, LLMs effectively handle time-related queries, such as “last week” or “during the 2012 Olympics,” enabling both historical and real-time data analysis. However, direct access to live data often depends on external integrations, like Geode. By combining their understanding of spatial relationships, location, and time, LLMs excel at reducing ambiguity in natural language queries, transforming them into actionable insights and fostering natural, seamless interactions with complex geospatial systems, all while enhancing accessibility and usability.

C. Integration of multiple data modalities

Large Language Models (LLMs) are becoming increasingly skilled at integrating and interpreting data from diverse sources, such as satellite imagery, GIS data, environmental measurements, and unstructured text. This ability to handle multimodal data is essential in the geospatial domain, where information often comes in many forms. For instance, advanced models can process both text and images simultaneously, enabling tasks like identifying buildings taller than 20 stories in satellite images based on a user’s query. Models like EarthMarker showcase the potential of visual prompting, where the system focuses on specific regions of interest in remote sensing imagery to provide precise insights. Systems like Geode enhance this multimodal approach by leveraging specialised components to manage diverse data types. Retrieval-augmented experts combine language understanding with retrieval methods to access geospatial data, improving reliability and reducing hallucinations. Task-specific models, fine-tuned on specific datasets, excel in areas like predicting rainfall or traffic patterns. Database experts access real-time and historical geospatial data, including weather, census, and geography, to ensure up-to-date information. Additionally, functional experts handle complex calculations, such as geometry and analytics, for more sophisticated analysis. By combining these specialised tools, LLMs can extract, interpret, and integrate multimodal geospatial data, delivering comprehensive and accurate responses. This growing capability allows users to interact with complex geospatial systems more naturally and make informed decisions based on data from multiple sources, demonstrating the transformative potential of LLMs in the geospatial field.

D. Zero-shot and few-shot learning capabilities in geospatial contexts

Large Language Models (LLMs) excel at geospatial tasks, even in a zero-shot setting, where they rely solely on pre-existing knowledge without prior examples. Success in this context depends on carefully crafting prompts to include all relevant information, such as database schemas, to guide the model effectively. Geode is an example of a system that handles zero-shot geospatial queries by integrating diverse data modalities. LLMs also perform well in few-shot learning scenarios, where they are given a small number of examples to identify patterns and generalize to new tasks. This in-context learning method is particularly useful when training data is limited, enabling models to generate accurate responses based on just a few illustrative examples. Optimizing prompts is critical to enhancing few-shot learning, with strategies such as advanced sampling, schema-based example selection, and leveraging external knowledge. For instance, examples can be chosen based on semantic similarity, diversity, relevance to the schema, or difficulty. Techniques like extracting a "question skeleton" from a query can further improve the selection of relevant examples. Cross-domain few-shot learning, where models generalize across different domains using limited examples, is another powerful approach, allowing LLMs to adapt to varied use cases without task-specific training. By leveraging zero-shot and few-shot learning with effective prompt optimization, LLMs can tackle complex geospatial tasks with minimal input, making them versatile tools for a wide range of domains.

IV. LLMS FOR GEOSPATIAL QUERY UNDERSTANDING

LLMs are increasingly being employed to generate structured queries for geospatial databases and code for Geographic Information Systems (GIS) operations, allowing users to interact with complex systems using natural language. This capability is transforming how users access and analyse geospatial data, moving beyond the traditional need for specialised knowledge of GIS tools and programming.

A. SQL Generation from Natural Language

Schema linking is crucial for text-to-SQL tasks, as it helps LLMs map natural language elements to database tables and columns, ensuring accurate SQL queries. LLMs often struggle with this when schema information is presented in plain text, needing to identify relevant tables even when they're implied. Techniques like Graph Neural Networks (GNNs), Chain-of-Thought prompting, and fine-tuning improve schema linking by refining entity connections and addressing overlooked join relations. Cross-domain generalization is another challenge, as LLMs must handle new, unseen databases, but real-world queries often involve synonyms or incomplete instructions, complicating the task. To enhance SQL generation, frameworks like Knowledge-to-SQL use Data Expert LLMs (DELLMs) to provide knowledge missing from the database schema. Structure-guided generation, such as with SGU-SQL, applies semantic-enhanced matching to guide SQL creation. Prompt engineering helps guide LLMs by crafting questions that maximize model understanding. Chain-of-thought prompting encourages LLMs to articulate their reasoning step-by-step, improving accuracy. Fine-tuning adapts the model for text-to-SQL tasks by training it on relevant datasets. Multi-agent collaboration, like MAC-SQL, uses multiple models to tackle complex queries. These methods, together, improve the flexibility and accuracy of text-to-SQL tasks, making database interactions more intuitive. Datasets like Spider, BIRD, and CoSQL provide large, cross-domain benchmarks for testing SQL generation, each with their own challenges, such as noisy data and complex schema contexts. These datasets help refine models by simulating real-world scenarios and pushing their ability to handle diverse SQL queries.

B. OverpassQL Generation for OpenStreetMap

LLMs are being increasingly used to generate Overpass Query Language (OverpassQL) queries from natural language, making it easier for users to access and analyze data from OpenStreetMap (OSM) without needing to learn the OverpassQL syntax. Research in this area includes the development of the Text-to-OverpassQL task, which focuses on translating complex natural language requests into executable OverpassQL code. A key contribution to this research is the creation of the OverpassNL dataset, a benchmark that contains over 8,500 real-world Overpass queries paired with corresponding natural language inputs. The task's evaluation considers various metrics, such as surface string similarity, semantic and syntax accuracy, and how well the generated queries perform when executed against the OSM database. Both sequence-to-sequence models and large language models, like GPT-4 with few-shot learning, have been used to tackle this challenge. By applying these models, users can seamlessly translate natural language queries into the technical language needed to extract specific data from OSM, streamlining the process of working with geographic data.

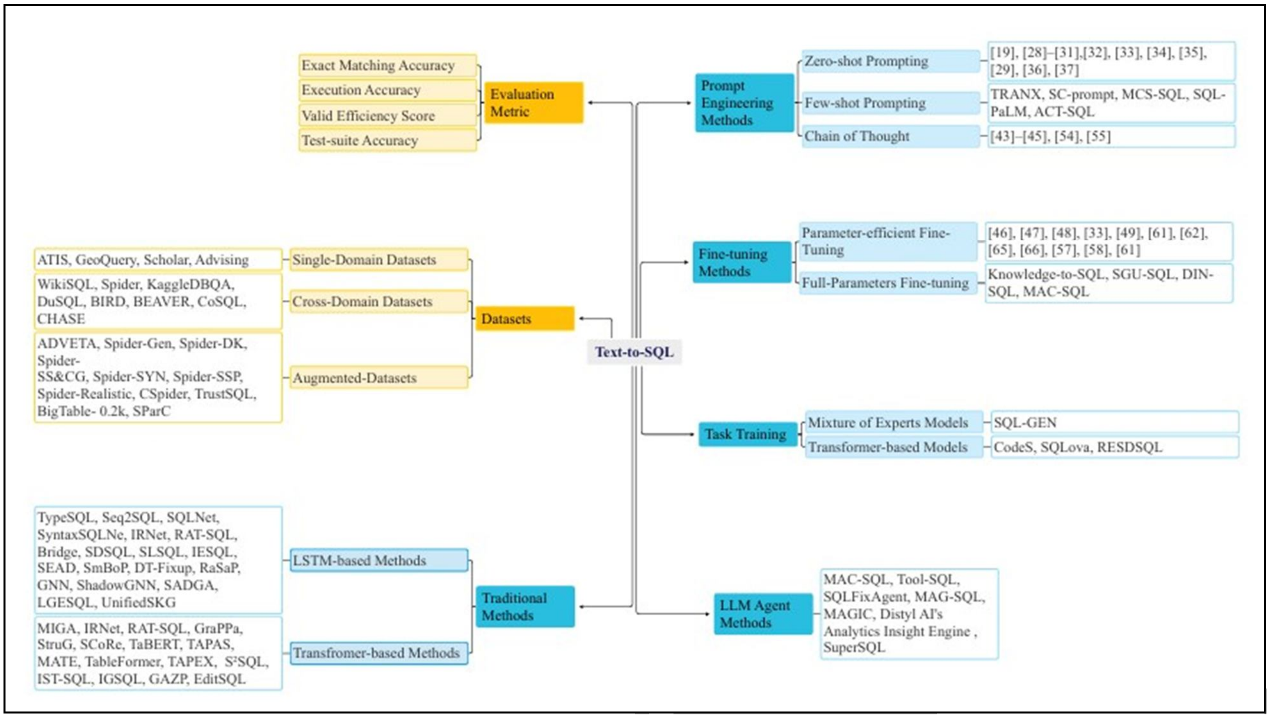
Exact Matching Accuracy
Execution Accuracy
Valid Efficiency Score
Test-suite Accuracy
Evaluation Metric
Prompt Engineering Methods
Zero-shot Prompting
[19], [28]–[31],[32], [33], [34], [35], [29], [36], [37]
Few-shot Prompting
TRANX, SC-prompt, MCS-SQL, SQL-PaLM, ACT-SQL
Chain of Thought
[43]–[45], [54], [55]
ATIS, GeoQuery, Scholar, Advising
Single-Domain Datasets
WikiSQL, Spider, KaggleDBQA, DuSQL, BIRD, BEAVER, CoSQL, CHASE
Cross-Domain Datasets
ADVETA, Spider-Gen, Spider-DK, Spider-SS&CG, Spider-SYN, Spider-SSP, Spider-Realistic, CSpider, TrustSQL, BigTable- 0.2k, SParC
Augmented-Datasets
Datasets
Fine-tuning Methods
Parameter-efficient Fine-Tuning
[46], [47], [48], [33], [49], [61], [62], [65], [66], [57], [58], [61]
Full-Parameters Fine-tuning
Knowledge-to-SQL, SGU-SQL, DIN-SQL, MAC-SQL
Text-to-SQL
Task Training
Mixture of Experts Models
SQL-GEN
Transformer-based Models
CodeS, SQLova, RESDSQL
TypeSQL, Seq2SQL, SQLNet, SyntaxSQLNe, IRNet, RAT-SQL, Bridge, SDSQL, SLSQL, IESQL, SEAD, SmBoP, DT-Fixup, RaSaP, GNN, ShadowGNN, SADGA, LGESQL, UnifiedSKG
LSTM-based Methods
MIGA, IRNet, RAT-SQL, GraPPa, StruG, SCoRe, TaBERT, TAPAS, MATE, TableFormer, TAPEX, S²SQL, IST-SQL, IGSQL, GAZP, EditSQL
Transfromer-based Methods
Traditional Methods
LLM Agent Methods
MAC-SQL, Tool-SQL, SQLFixAgent, MAG-SQL, MAGIC, Distyl AI's Analytics Insight Engine , SuperSQL

Fig 1. Overview of Text-to-SQL Methodology

C. Code Generation for GIS Tasks (e.g., PyQGIS):

Large language models (LLMs) are increasingly being used to generate code for Geographic Information System (GIS) tasks, especially with tools like PyQGIS, the Python API for QGIS, allowing users without coding skills to perform complex geospatial analyses through natural language. Key to this progress is fine-tuning LLMs on geospatial datasets, helping models grasp geospatial concepts and their relationships, which reduces errors. Techniques like Named Entity Recognition (NER) and ontology mapping are essential for understanding geospatial terms in queries, and systems like ChatGeoAI use these methods to translate natural language into executable PyQGIS code. These systems can handle a variety of tasks, from basic mapping to more advanced spatial analysis, such as routing and network analysis. To improve accuracy, some systems incorporate a library of pre-defined code snippets, which serve as reliable building blocks, reducing complexity. Additionally, developing self-correcting mechanisms can enhance performance, allowing models to learn from mistakes and refine their code output. Although current systems may still struggle with ambiguous attribute names or aliases, the integration of LLMs into GIS makes geospatial analysis more accessible and user-friendly. Hybrid approaches are also showing promise, combining LLMs' natural language capabilities with traditional GIS tools for better data processing and analysis. This flexibility opens the door to more efficient, intuitive geospatial workflows.

V. METHODOLOGIES AND TECHNOLOGIES

A. Prompt Engineering:

This is a critical aspect of using LLMs for geospatial queries, as the design of the prompt significantly impacts the accuracy of the generated outputs.

1) Importance of Prompt Design:

Effective prompt design is essential for guiding Large Language Models (LLMs) to generate accurate SQL queries. It involves providing the right context, such as database schemas, clear instructions, and relevant examples. The way the schema is presented—such as using "CREATE TABLE" SQL statements with column types and primary/foreign keys—greatly impacts accuracy. Studies show that more database content in prompts can sometimes reduce accuracy, emphasizing the sensitivity of LLMs to input information. Including database-related knowledge and structuring prompts carefully is key. A comprehensive evaluation highlighted that aspects like question representation, example selection, and organization play a major role in performance, pointing to the need for a systematic approach to prompt engineering. Techniques like in-context learning, where a few examples are provided, allow LLMs to apply learned patterns to new queries without specific task training. Few-shot prompting, especially with diverse yet similar examples to the user's query, can significantly boost performance. SQL-PaLM, for example, shows that concise prompts, using symbols instead of natural language for schema description, improve clarity. Additionally, how prompts linearize structured knowledge—whether as "text" or "code"—can influence results, including key details like data types and foreign key constraints. Chain-of-Thought (CoT) prompting, where the model breaks down its reasoning step-by-step, can further improve accuracy, though it may require adaptation for text-to-SQL tasks. In summary, prompt design is crucial for LLMs' success in text-to-SQL tasks and must be thoughtfully crafted with the right context and structure.

2) Strategies

Few-shot examples are essential for boosting the performance of Large Language Models (LLMs) in text-to-SQL tasks, as they help the model learn patterns and generate accurate responses. The selection of examples is crucial, with both similarity and diversity playing key roles in improving results. Presenting examples based on the SQL query structure or including full details like instructions, schema, questions, and SQL queries can enhance effectiveness. For large models, providing just the question-SQL pair may suffice. Including structured database schemas—such as table names, column types, and relationships like foreign keys—within the prompt helps LLMs generate more accurate SQL.

Studies show that presenting schemas as "CREATE TABLE" SQL statements, including all necessary details, improves prediction accuracy. Normalized schemas tend to work better than unnormalized ones, and database content represented through specific prompts like "SelectRow" or "SelectCol" can further boost performance. Additionally, Chain of Thought (CoT) prompting guides LLMs through a step-by-step reasoning process, which enhances accuracy by incorporating intermediate steps. While CoT may need adjustments for text-to-SQL tasks, its ability to break down problems and use results from each step has proven effective. Self-correction techniques that rely on these intermediate steps also contribute to better outcomes. Overall, well-chosen examples, structured database information, and thoughtful prompting strategies like CoT are vital for improving the accuracy and efficiency of LLMs in generating SQL queries.

3) In-context Learning

In-context learning (ICL) is crucial for improving the performance of Large Language Models (LLMs) in text-to-SQL tasks, as it allows models to learn from examples included in the prompt. The selection and organization of these examples are key to ICL's effectiveness. Research shows that using the syntactic structure of SQL queries transforming them into discrete feature vectors helps identify patterns and enhances comparison. Approaches like similarity-based sampling (choosing semantically similar examples), diversity-based sampling (selecting varied examples), and hybrid sampling (combining both methods) are commonly used to maximize learning. Studies highlight that ICL-based methods consistently outperform other approaches, leading to higher accuracy and lower computational costs. Benchmarking studies further validate ICL's importance, showing its role in evaluating LLMs and refining prompt engineering strategies. The organization of examples in the prompt also affects performance whether it's full information (including instructions, schema, and SQL queries), just SQL queries, or question-SQL pairs. While question-SQL pairs work well for powerful models, providing full examples can be more effective for others. These evaluations help optimize example selection, improving the semantic understanding and efficiency of LLMs, ultimately contributing to better text-to-SQL systems.

B. Fine-Tuning:

Fine-tuning pre-trained Large Language Models (LLMs) with geospatial data and task-specific information is key to enhancing their performance across a range of applications. For example, the "ChatGeoAI" system fine-tunes the Llama 2 model using a dataset of natural language prompts and PyQGIS code, incorporating domain-specific ontologies and named entity recognition (NER) to improve geospatial understanding. Similarly, the "GeoLLM" method uses map data from OpenStreetMap to fine-tune LLMs, unlocking their geospatial knowledge and enhancing their performance by incorporating nearby location information. For text-to-SQL tasks, fine-tuning with database schemas and natural language question-SQL pairs, as seen in the "SQL-PaLM" framework, helps LLMs better understand SQL syntax and generate accurate queries. Fine-tuning also improves performance on sub-tasks within text-to-SQL tasks and can be further enhanced by expanding data coverage, using synthetic data, and integrating query-specific content. Parameter-efficient techniques like Low-Rank Adaptation (LoRA) reduce the computational cost and memory requirements of fine-tuning by adjusting fewer parameters, which increases efficiency without compromising performance. Studies also show that larger models, when well-aligned with their training data, require less data for fine-tuning. Overall, fine-tuning LLMs with domain-specific data and using techniques like LoRA enables LLMs to generate more accurate, contextually relevant outputs, making them highly effective for specialized tasks like geospatial analysis and SQL query generation.

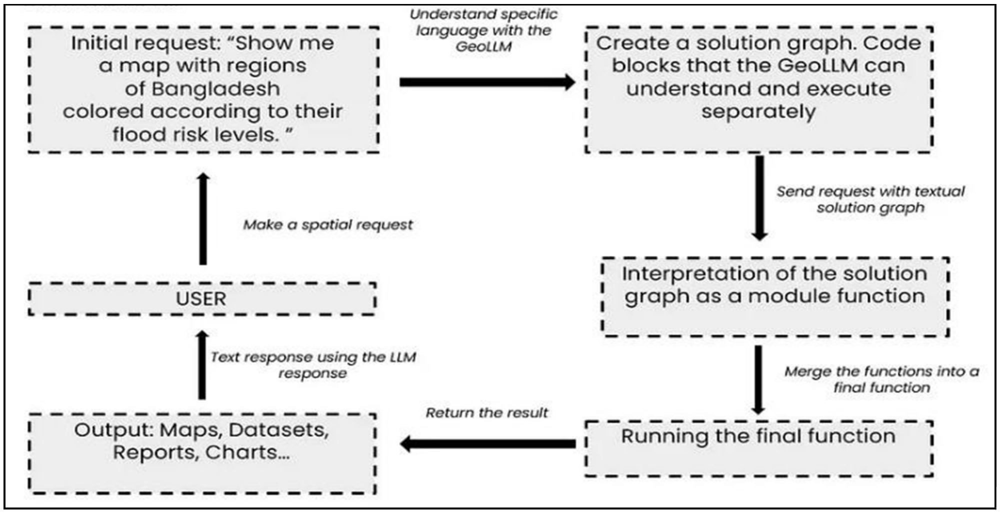
Initial request: "Show me a map with regions of Bangladesh colored according to their flood risk levels."
Understand specific language with the GeoLLM
Create a solution graph. Code blocks that the GeoLLM can understand and execute separately
Make a spatial request
Send request with textual solution graph
USER
Interpretation of the solution graph as a module function
Text response using the LLM response
Merge the functions into a final function
Return the result
Output: Maps, Datasets, Reports, Charts...
Running the final function

Fig 2. Workflow Overview of Geospatial Retrieval using LLMs

C. Data Augmentation

Increasing training data and enhancing its diversity through data augmentation are key to improving the performance of Large Language Models (LLMs), especially in specialized areas like geospatial analysis and text-to-SQL tasks. Data augmentation helps models generalize better and perform more effectively by artificially expanding training datasets. In the "SQL-PaLM" framework, synthetic data is used to generate new training examples, expanding the dataset and helping the model handle complex and diverse inputs.

This can include using LLMs to create synthetic SQL queries or employing rule-based generation for better control and precision. Other techniques like bi-directional generation, error-based expansion, and data transformation (e.g., paraphrasing or schema randomization) also contribute to creating more varied and robust datasets. Research shows that training on more diverse datasets improves model performance, with models trained on combined datasets like BIRD and Spider outperforming those trained on a single dataset. Data augmentation not only increases the quantity of data but also addresses potential gaps or errors in existing data, reinforcing the model's knowledge. This process helps mitigate issues like overfitting and reduces data scarcity, especially in fields where high-quality labeled data is hard to come by. Ultimately, data augmentation is a powerful tool for improving model robustness, generalization, and accuracy in real-world applications, making LLMs more capable of handling a wide range of tasks effectively.

VI. DATA-SETS AND EVALUATION METRICS

Several datasets and benchmarks are crucial for training and evaluating the performance of LLMs in geospatial and database query tasks. These datasets cover a range of complexities and focus areas, providing a foundation for the development of more robust and accurate natural language interfaces.

A. Data - Sets

1) OverpassNL

This dataset is specifically designed for the Text-to-OverpassQL task, which involves generating Overpass queries from natural language inputs. It contains 8,352 real-world Overpass queries collected from the OpenStreetMap (OSM) community, paired with natural language descriptions written by trained annotators. The queries cover a wide range of OverpassQL syntax features and have high geographical coverage. The dataset is split into training (6,352 instances), development (1,000 instances), and test (1,000 instances) sets, with no duplicates between training and evaluation sets. Each query has an average of 11.9 syntactic units, and the average length of a query is 199.8 characters. The queries are based on actual information needs of users. OverpassNL aims to support the full functionality of the Overpass Query Language without simplification. The Overpass Query Language (OverpassQL) is used to extract information from the OpenStreetMap database.

2) Spider

Spider is a large-scale, cross-domain dataset for Text-to-SQL tasks, featuring 8,659 question-SQL pairs in its training set and 1,034 in the development set, spanning 200 databases and 138 domains. Known for its complexity, Spider is considered one of the most challenging and influential benchmarks in the field of text-to-SQL. There are several variations of the Spider dataset that aim to test different aspects of model performance: Spider-Realistic, a subset of the development set, revises questions by removing explicit references to column names to focus on generalization; Spider-Syn replaces schema-related terms with manually chosen synonyms to assess robustness to vocabulary changes; and Spider-DK incorporates domain knowledge into the question-SQL pairs to test models' ability to leverage external information.

*3) BIRD (*Big *Bench for large-scale Database Grounded text-to-SQL Evaluation)*

This is a challenging, **cross-**domain dataset for Text-to-SQL that emphasizes real-world database content and external knowledge. BIRD includes over 12,751 question-SQL pairs across 95 large databases covering over 37 professional domains. It contains 9,428 training and 1,534 development question-SQL pairs. The dataset focuses on the complexity of the database, the need for external knowledge, and SQL query efficiency. BIRD is a leading benchmark focused on massive and real database content, introducing knowledge reasoning between natural language questions and database content.

4) TCQL

The TCQL dataset is an important resource for advancing research in the text-to-CQL task, which focuses on translating natural language into Corpus Query Language (CQL). CQL is a specialized query language for analyzing linguistically annotated text corpora, enabling complex searches based on linguistic features. The dataset was created due to the lack of dedicated resources for text-to-CQL tasks, unlike text-to-SQL, which has more established datasets. To construct the TCQL dataset, collocation extraction techniques were used to generate candidate CQL queries, followed by manual annotation by experts and a multi-stage labeling process involving GPT-4 and human review. The dataset is divided into three categories—simple, within, and condition—based on the complexity of the CQL queries. It's used to evaluate state-of-the-art models, incorporating novel metrics like CQLBLEU to assess syntactic and semantic accuracy. The TCQL dataset plays a crucial role in developing systems that bridge the gap between natural language descriptions and the intricate syntax of CQL, providing a benchmark for evaluating LLMs and fostering advancements in corpus development, linguistics, and NLP.

B. Evaluation Metrics

Evaluation metrics for LLMs in geospatial and text-to-SQL tasks fall into a few main categories, measuring different aspects of performance.

1) Exact Match (EM)

Exact Match (EM) is a metric used to evaluate the syntactic correctness of generated queries by checking if a query exactly matches a reference query, with no differences. In both text-to-SQL and text-to-CQL tasks, EM measures whether the predicted query is identical to the ground truth, including all components and sequence. However, EM is a strict metric, which can penalize semantically correct queries if they do not match the reference exactly. For example, in text-to-SQL, different syntactically correct queries that produce the same result may not be considered valid by EM, leading to false negatives. Similarly, in text-to-CQL, EM can fail to account for semantically equivalent but syntactically different queries. While useful for syntactic evaluation, EM doesn’t consider semantic equivalence, meaning functionally correct queries can be marked as incorrect if they don’t match the reference syntax. EM also doesn’t account for nuances where multiple valid queries exist for a single question and is too strict in requiring exact matches. It also doesn’t address value differences, which can cause false positives and negatives. To overcome these limitations, EM is often combined with other metrics like Execution Accuracy (EX), which evaluates the correctness of a query based on its execution result, helping to assess the semantic validity of the query even if its syntax differs from the reference.

2) Valid Efficiency Score (VES)

The Valid Efficiency Score (VES) is a key evaluation metric in the text-to-SQL domain, especially for the BIRD dataset, that measures both the accuracy and efficiency of generated SQL queries. It combines Execution Accuracy (EX), which checks if a generated query returns the same results as the ground truth, and execution efficiency, which looks at how quickly the query runs. VES penalizes correct but inefficient queries, such as those with redundant joins or subqueries that increase execution time, ensuring models generate SQL queries that are both accurate and optimized. This metric is especially important in large, complex databases where performance matters. The VES score is calculated by averaging the product of an indicator function (which is 1 if the results match and 0 if they don’t) and the relative execution efficiency, the latter being the square root of the ratio of execution times for the ground truth and generated queries.The BIRD benchmark also stabilizes VES by averaging execution efficiency over 100 runs, reducing the impact of database performance variability. This makes VES a more reliable metric for real-world applications, where correctness and speed are both essential. Some studies may also use Correct-VES (C-VES) to focus specifically on the efficiency of correctly generated queries. VES is vital for evaluating models in practical scenarios, ensuring both accuracy and resource efficiency in SQL generation.

3) Test-suite Accuracy *(TS)*

Test Suite Accuracy (TS) is a metric in the text-to-SQL domain that evaluates the robustness of generated SQL queries by testing them across multiple, augmented database versions. Unlike Execution Accuracy (EX), which checks if a query matches the ground truth on a single database, TS ensures that queries can handle variations in data while still returning correct results. This is achieved through database augmentation, where different data sets with the same schema are used to test queries. For a query to pass TS, it must successfully execute on all augmented database versions, making it a more rigorous and comprehensive evaluation compared to EX. TS also reduces the risk of false positives, which can occur when a query works on one database but fails on others. By testing across multiple scenarios, it assesses not only correctness but also whether the query is generalized enough to handle diverse data, making it more reliable in real-world situations. TS helps prevent models from overfitting to a specific database, ensuring they perform well across different data versions. This metric is valuable because it aligns with practical conditions where database variations are common, and a high TS score indicates that a model is likely to generate reliable queries in varied scenarios. Ultimately, Test Suite Accuracy is a more thorough metric than EX, ensuring that text-to-SQL systems are both accurate and robust for real-world use.

4) Semantic Similarity Metrics

Key Value Similarity (KVS) is a vital metric for evaluating Text-to-OverpassQL systems, as it measures how closely a generated query aligns with a reference query in terms of key-value pairs. The metric works by identifying the key-value pairs in both the generated query (qG) and the reference query (qR), then calculating the intersection of these pairs and normalizing it by the size of the larger set. This focus on key-value pairs allows KVS to evaluate semantic relatedness, disregarding differences in syntax or structure that don't affect meaning. A higher KVS score indicates a stronger semantic match between the two queries. Unlike surface-level string similarity metrics, KVS emphasizes the core semantic content, making it particularly useful for the Text-to-OverpassQL task. It is part of the Overpass Query Similarity (OQS) metric, which also includes a character-level F-score (chrF) and syntactic tree similarity (TreeS), providing a comprehensive evaluation of both semantic and syntactic aspects. While metrics like Component Matching (CM) are used for evaluating text-to-SQL queries based on SQL components, KVS is more suited for evaluating Text-to-OverpassQL, where queries rely on key-value pairs rather than SQL syntax. Overall, KVS plays a crucial role in measuring the semantic similarity between generated and reference queries, ensuring more accurate evaluations in Text-to-OverpassQL systems.

VII. CHALLENGES AND LIMITATIONS

While Large Language Models (LLMs) offer significant potential for advancing geospatial analysis, several challenges and limitations need to be addressed to ensure their effective and reliable deployment. These issues span the complexity of geospatial data, the inherent ambiguity of natural language, and the computational demands of LLM-based systems, among others.

A. Ambiguity in Natural Language

Natural language queries often include complex structures such as nested clauses, pronouns, and vague terms, making it challenging for LLMs to accurately interpret the user's intent. The inherent ambiguity of natural language, with multiple possible interpretations for a given question, further complicates the task. User queries can include synonyms, typos, vague expressions, and incomplete instructions, all of which hinder accurate interpretation. Furthermore, word sense ambiguity (where a word has multiple meanings) and word segmentation ambiguity (where words are not clearly separated) add to the challenge.

B. Need for Domain-Specific Knowledge

LLMs often lack the specific geographic knowledge and understanding of geospatial concepts needed for accurate analysis. They can struggle with interpreting numerical representations such as latitude and longitude in relation to real-world locations. Moreover, geospatial terminology and operations (e.g., buffer, clip, intersect) are often not well represented in general-purpose models. LLMs need to understand how to interpret specific data types or formats in a database and apply unique rules and regulations when generating SQL or geospatial code.

C. Hallucination and its Detection/Mitigation

A significant challenge with LLMs is their tendency to generate incorrect information or code that deviates from user intent, factual knowledge, or context – a phenomenon known as hallucination. These hallucinations can manifest in various forms including input conflicts, context conflicts, factual conflicts, intent conflicts, inconsistencies, and knowledge conflicts. The consequences of such hallucinations can include incorrect code, misleading results, reduced user trust, and decreased efficiency. Effective strategies for mitigating hallucinations include self-refinement techniques, execution-aware methods, test case generation, dedicated detection tools, and data augmentation.

D. Evaluating the Quality of Generated Queries and Code:

Developing suitable evaluation metrics for assessing the quality of generated queries and code remains a considerable challenge. Existing benchmarks and datasets may not fully capture the nuanced requirements of geospatial applications. Evaluation must consider the complexity of the generated SQL, the correctness of the query syntax, the efficiency of the code, and the relevance to user intent. There is a need for metrics that assess semantic similarity alongside syntactic accuracy.

E. Complexity of Geospatial Data:

Geographic data is challenging due to its complexity and abundance. Master's degrees require processing of various types of data such as satellite imagery, GIS data, and environmental measurements, each with its own unique structure and meaning. Adding to the complexity, geographic data often covers long periods of time, so LLM.

Students need to address spatio-temporal questions to understand how things change over time and space. How close the nodes are, whether they are in different locations, or whether they share a border adds to the complexity. Moreover, the volume of geographic data is very large and needs to be optimized, especially when dealing with large datasets or complex queries. To solve a complex problem, LLM also needs to understand the connections between multiple tables and each row in that data, adding another layer of complexity.

VIII. CONCLUSION

In conclusion, LLMs are proving to be a powerful tool for geospatial analysis. With continued research and development, these technologies hold the potential to transform how users interact with and derive insights from geospatial data, making it more accessible, efficient, and actionable across various domains. The future of geospatial analysis will likely involve hybrid systems combining the strengths of LLMs with traditional GIS tools, enhanced with tailored evaluation metrics and continuous refinement, pushing the boundaries of both academic research and practical application.

REFERENCES

[1] Staniek, M., Schumann, R., Zufle, M., & Riezler, S. (2023). Text-to-OverpassQL: A Natural Language Interface for Complex Geodata Querying of OpenStreetMap. arXiv.Org, abs/2308.16060.

[2] Roberts, J., Luddecke, T. O. D., Das, S., Han, K., & Albanie, S. (2023). GPT4GEO: How a Language Model Sees the World's Geography. arXiv.Org, abs/2306.00020.

[3] Manvi, R., Khanna, S., Mai, G., Burke, M., Lobell, D. B., & Ermon, S. (2023). GeoLLM: Extracting Geospatial Knowledge from Large Language Models. arXiv.Org, abs/2310.06213.

[4] Gupta, D. V., Ishaqui, A. S. A., & Kadiyala, D. K. (2024). Geode: A Zero-shot Geospatial Question-Answering Agent with Explicit Reasoning and Precise Spatio-Temporal Retrieval. arXiv.Org, abs/2407.11014.

[5] Fakhoury, S., Naik, A., Sakkas, G., Chakraborty, S., & Lahiri, S. K. (2024). LLM-based Test-driven Interactive Code Generation: User Study and Empirical Evaluation. IEEE Transactions on Software Engineering, 1–15.

[6] Zhang, T., Chen, C., Liao, C., Wang, J., Zhao, X., Yu, H., Wang, J., Li, J., & Shi, W. (2024). SQLfuse: Enhancing Text-to-SQL Performance through Comprehensive LLM Synergy. arXiv.Org, abs/2407.14568.

[7] Ning, H., Li, Z., Akinboyewa, T., & Lessani, M. N. (2024). LLM-Find: An Autonomous GIS Agent Framework for Geospatial Data Retrieval. arXiv.Org, abs/2407.21024.

[8] Sarker, S. D., Dong, X., Li, X., & Qian, L. (2024). Enhancing LLM Fine-tuning for Text-to-SQLs by SQL Quality Measurement.

[9] Zhu, X., Li, Q., Cui, L., & Liu, Y. (2024). Large Language Model Enhanced Text-to-SQL Generation: A Survey.

[10] Zhang, B., Ye, Y., Du, G., Hu, X., Li, Z., Yang, S., Liu, C. H., Zhao, R., Li, Z., & Mao, H. (2024). Benchmarking the Text-to-SQL Capability of Large Language Models: A Comprehensive Evaluation. arXiv.Org, abs/2403.02951.

[11] Roberts, J., Luddecke, T. O. D., Das, S., Han, K., & Albanie, S. (2023). GPT4GEO: How a Language Model Sees the World's Geography. arXiv.Org, abs/2306.00020.

[12] Gao, D., Wang, H., Li, Y., Sun, X., Qian, Y., Ding, B., & Zhou, J. (2024). Text-to-SQL Empowered by Large Language Models: A Benchmark Evaluation. Proceedings of The Vldb Endowment, 17(5), 1132–1145.